REMARKS

Favorable reconsideration of the application is respectfully requested in light of the amendments and remarks herein.

Upon entry of this amendment, claims 1-9 and 11 will be pending. By this amendment, claim 10 has been canceled; and claims 1, 9, and 11 have been amended. No new matter has been added.

Objections to the Drawings

In Section 1 of the Office Action, the drawings are objected to for failing to be in compliance with 37 C.F.R §1.121(d). New replacement drawings in compliance with 37 C.F.R §1.121(d) are submitted herein in the attached Appendix.

In Figure 2, legend 14 designating the transport stream has been changed to legend 15. Furthermore, corresponding paragraphs in the Specification referencing the transport stream of Figure 2 has been amended.

§112 Rejection of Claim 10

In Section 2 of the Office Action, claim 10 stands rejected under 35 U.S.C. §112, second paragraph. Claim 10 has been canceled.

§102 Rejection of Claims 1-3 and 7-11

In Section 4 of the Office Action, claims 1-3 and 7-11 stand rejected under 35 U.S.C. §102(b) as being anticipated by McClellan (U.S. Patent No. 5,619,250). Claims 1, 9, and 11 have been amended to clarify the meaning of the terms in the claims.

In the Background section of the Specification, it was disclosed that "[i]t is known to transmit data objects in a transport stream for downloading by an end user. The objects can include any form of data for use by the end user and can be transmitted in a distributed repetitive form, for instance by means of a data carousel. Often, data objects may be grouped together as data modules such that, for an application of an end user to make use of a particular object, the entire module in which it is contained has to be downloaded from the transport stream. In these circumstances, service providers may group together related objects in respective modules such that, if an end user stores a downloaded module, the access time for the related objects is greatly reduced." *Background of the Specification, page 1, lines 14-22.*

"Under these circumstances, it is desirable that, when an application of an end user requests an object, it should receive the latest version of that object. However, if, before the object was updated, the end user had already downloaded the respective module, the application will be provided with the old version of the object as contained in the modules stored by the end user." *Background of the Specification, page 1, line 30 to page 2, line 3.*

To address the above-described problem, embodiments of the present invention provide device, method, and computer program for providing downloaded objects to an application. For example, the structure of device claim 1, as presented herein, includes:

> "*an object layer interface* for downloading from a received transport stream a module containing at least one object including an object requested by an application;
>
> *a module memory* for storing the downloaded module;
>
> *a controller* for monitoring whether the module in the module memory is a current version, wherein
>
> when the application requests an object contained in a stored module and the stored module is a current version, the object layer interface provides the application with the object from the module memory and,

when the application requests an object contained in a stored module and the stored module is not a current version, the object layer interface downloads the current version of the module and overwrites the module stored in the module memory; and

an object cache for storing objects, wherein

when the application requests an object contained in a stored module and the stored module is not the current version, the controller copies into the object cache only the objects of the stored module still in use by the application before the object layer interface overwrites the stored module with the current version of the stored module."

(emphasis added)

As explained above, downloadable objects are provided in modules, where each module contains at least one object. When a particular object is requested to be downloaded, the module which includes that object will be downloaded. However, in the process of downloading the module, all of the other objects in the same module will also be downloaded. Thus, when the latest version of a particular object is requested by an application, that object along with all of the other objects in the downloaded module will be received. The additional objects which were not requested can be discarded but that would be wasteful. Therefore, the additional objects can be stored so that if any of those other objects are requested subsequently, they will be available immediately.

The problem with storing all of the objects of the downloaded module is that the application may already be using an earlier version of one of those other objects. In general, it would not be possible to overwrite the earlier version of an object with a new version of that object while the object was being used by the application. This would cause a malfunction of the application. Thus, the problem with downloading and storing an entire module is that, in addition to the requested new version of an object, there may be one or more other objects in the module currently being used by the application.

The embodiments of the present invention address the above-described problem by copying and caching objects of the module still in use by the application. This allows the application to continue to use any objects which it is already using while all of the objects of a downloaded module are stored for future use. If the application then requests one of the additional objects, the application will receive the most up-to-date version. Thus, if the application stops using a particular object and requests that object again, the application receives and is able to use the latest version of the object as downloaded with the module rather than the previous version of the object copied during downloading of the new module.

By contrast, McClellan is directed to a set top box having an operating system which comprises "modules" of code. The term "module" in McClellan is used to indicate that the code has a modular structure with each "module" making up a portion or component of the overall system. Thus, the "modules" of McClellan are not comparable to the modules of the present invention. Further, McClellan fails to disclose that its "modules" include objects.

Although the Examiner refers to column 9, line 35 to column 10, line 14 and column 10, lines 54-55 in arguing that the "description record" of McClellan anticipates the objects of the present invention, it is respectfully submitted that that the term "object" has a well-known meaning in the art and that the description records cannot be interpreted as being "objects" as used in the related art. For example, claim 1 defines a device for providing downloaded "objects to an application" but McClellan does not teach providing "description records" to an application in the same context. Claim 1 defines an "objects layer interface" but there is no teaching of a "description record layer interface" in McClellan. Claim 1 refers to the application requesting an object and providing it to the application but McClellan provides no teaching of this. Claim 1 defines an “object cache” for storing objects but McClellan fails to disclose a storage for objects.

The Examiner refers to a description of the ROM in column 6, lines 57-65 in indicating that McClellan discloses a storage for objects. However, the ROM of McClellan is not used for copying “description records” into it when respective modules are being overwritten and those “description records” are still in use by the application. Further, column 6, lines 57-65 of McClellan indicates the ROM cannot be used as an object cache (which is required in claim 1) when it states that the ROM stores modules rather than objects, and that modules “cannot be removed” (i.e., “[s]imilarly, new modules cannot be added to the ROM because it is not possible to write to ROM”).

McClellan does describe the possibility of downloading “modules” as individual units. However, with McClellan, there is no desire to use only portions of those “modules”. Further, there is no situation in which part of a “module” is already being used.

McClellan addresses a problem that arises when an application program is loaded into the set top box, the application may find that a “module” of the operating system is missing or outdated (see column 7, lines 6-22). If the operating system is found to be insufficient, then a request is made for a new “module” to be downloaded (see column , line 23-35). Therefore, for the system described in McClellan, the application program is configured to wait until the required “module(s)” is/are downloaded. This is in contrast to the present invention, where an application may already be running and using objects contained within a new module required for downloading.

In conclusion, McClellan fails to teach or suggest the possibility of an application using part of a “module” and needing to continue to use that part while downloading a new version of the same module. There is then no suggestion of copying those parts of the module still in use while downloading a new version of the same module. McClellan further fails to teach or suggest overcoming the above-described problems by selectively storing objects currently in use in an object cache while the module to which those objects belong is overwritten in the module memory. The "description records" of McClellan cannot be considered to be objects described in the present invention. McClellan fails to teach or suggest a suitable memory for operating as an object cache.

Based on the foregoing discussion, it is maintained that claim 1, as presented herein, should be allowable over McClellan. Furthermore, since independent claims 9 and 11, as presented herein, closely parallel, and include substantially similar limitations, as independent claim 1, claims 9 and 11 should also be allowable over McClellan. Since claims 2-3 and 7-8 depend from claim 1, claims 2-3 and 7-8 should also be allowable over McClellan. Claim 10 has been canceled.

Accordingly, it is submitted that the Examiner's rejection of claims 1-3 and 7-11 based upon 35 U.S.C. §102(b) has been overcome by the present remarks and withdrawal thereof is respectfully requested.

§103 Rejection of Claims 4-6

In Section 6 of the Office Action, claims 4-6 stand rejected under 35 U.S.C. §103(a) as being unpatentable over McClellan in view of Koninklijke (WO 99/65230).

Based on the foregoing discussion regarding claim 1, and since claims 4-6 depend from claim 1, claims 4-6 should be allowable over McClellan. Further, it was stated that Koninklijke teaches a transmission system in a television network environment wherein interactive applications request and receive DSM-CC compliant modules having DSM-CC objects.

Therefore, McClellan and Koninklijke, individually or in combination, fail to teach or suggest all the limitations of claims 4-6.

Accordingly, it is submitted that the Examiner's rejection of claims 4-6 based upon 35 U.S.C. §103(a) has been overcome by the present remarks and withdrawal thereof is respectfully requested.

Conclusion

In view of the foregoing, entry of this amendment, and the allowance of this application with claims 1-9 and 11 are respectfully solicited.

In regard to the claims amended herein and throughout the prosecution of this application, it is submitted that these claims, as Originally presented, are patentably distinct over the prior art of record, and that these claims were in full compliance with the requirements of 35 U.S.C. §112. Changes that have been made to these claims were not made for the purpose of patentability within the meaning of 35 U.S.C. §§101, 102, 103 or 112. Rather, these changes were made simply for clarification and to round out the scope of protection to which Applicant is entitled.

In the event that additional cooperation in this case may be helpful to complete its prosecution, the Examiner is cordially invited to contact Applicant's representative at the telephone number written below.

The Commissioner is hereby authorized to charge any insufficient fees or credit any overpayment associated with the above-identified application to Deposit Account 50-0320.

Respectfully submitted,

FROMMER LAWRENCE & HAUG LLP

By: Samuel S. Lee, Reg. No. 42,791 for
William S. Frommer
Reg. No. 25,506
(212) 588-0800

Attachments